

## WEB-APPLICATION SERVER AND METHOD FOR SUPPLYING DYNAMIC CONTENTS THEREOF

**Publication number:** KR20030046911

**Publication date:** 2003-06-18

**Inventor:** KIM JEONG SUK (KR); LEE YUN JUN (KR); LIM MIN YEOL (KR); SONG IN GEUN (KR)

**Applicant:** LIM MIN YEOL (KR)

**Classification:**

- **International:** G06F17/00; G06F17/00; (IPC1-7): G06F17/00

- **European:**

**Application number:** KR20010077241 20011207

**Priority number(s):** KR20010077241 20011207

[Report a data error here](#)

### Abstract of KR20030046911

**PURPOSE:** A web-application server and a method for supplying dynamic contents thereof are provided to supply an execution result to a corresponding client by transmitting a web-application for creating dynamic contents to an objected server adjacent to a corresponding client. **CONSTITUTION:** In the case that a client requests contents, a request processing unit(100) executes a process according to kinds of the contents and transmits a response message to a corresponding client. A system monitor(307) monitors system information of a web-application server for judging whether the web-application server creates dynamic contents requested from clients. A web-application unit(400) stores a web-application and creates dynamic contents requested by the client. In the case that the web-application server does not create dynamic contents requested from clients by itself, a transmission manager(305) selects an objected server for a redirection of the current dynamic contents request out of many web-application servers based on external meta data for managing information of many web-application servers and information of the client who requested corresponding dynamic contents. A transmitting/receiving unit(300a) executes a data exchange with the web-application server selected as the objected server by the transmission manager(305).

---

Data supplied from the **esp@cenet** database - Worldwide

특2003-0046911

## (19) 대한민국특허청(KR)

## (12) 공개특허공보(A)

(51) Int. Cl.<sup>7</sup>  
G06F 17/00(11) 공개번호 특2003-0046911  
(43) 공개일자 2003년06월18일(21) 출원번호 10-2001-0077241  
(22) 출원일자 2001년12월07일

(71) 출원인 임민열

대전광역시 유성구 구성동 한국과학기술원 전기 및 전자공학과 3210호

(72) 발명자 임민열

대전광역시 유성구 구성동 한국과학기술원 전기 및 전자공학과 3210호

김정숙

서울특별시 강서구 가양동 가양아파트 B동 511호

송인근

대전광역시 동구 판암동 주공 104동 102호

이윤준

대전광역시 유성구 어은동 한빛아파트 101동 1304호

(74) 대리인 김함곤, 안광석, 박영일

## 설명 : 있음

## (54) 웹-애플리케이션 서버 및 그의 동적 컨텐츠 제공 방법

## 요약

본 발명은 웹-애플리케이션 서버 및 그의 동적 컨텐츠 제공 방법에 관한 것으로서, 클라이언트들이 요청한 컨텐츠를 제공하는 웹-애플리케이션 서버에 있어서, 클라이언트가 컨텐츠를 요청할 경우, 그 컨텐츠의 종류에 따른 처리를 수행하여 해당 클라이언트에게 응답 메시지를 전송하는 요청처리기와, 상기 웹-애플리케이션 서버가 클라이언트들로부터 요청된 동적 컨텐츠를 자체적으로 생성할 수 있는지의 여부를 판단하기 위해, 상기 웹-애플리케이션 서버의 시스템 정보를 모니터링하는 시스템 모니터와, 동적 컨텐츠들을 생성하기 위한 웹-애플리케이션을 내장하고, 클라이언트가 요청한 동적 컨텐츠를 생성하는 웹-애플리케이션부와, 상기 시스템 모니터의 모니터링 결과, 상기 웹-애플리케이션 서버가 클라이언트들로부터 요청된 동적 컨텐츠를 자체적으로 생성할 수 없다고 판단된 경우, 메타데이터 및 해당 동적 컨텐츠를 요청한 클라이언트의 정보에 의거하여, 동적 컨텐츠 요청을 리다이렉션(redirection)하기 위한 대상서버를 선정하는 전송매니저와, 상기 전송매니저에 의해 대상서버로 선정된 웹-애플리케이션 서버와의 데이터 교환을 수행하는 송/수신부를 포함하는 것을 특징으로 하며, 클라이언트의 요구에 의해 실시간으로 동적 컨텐츠를 생성하여야 하는 웹-애플리케이션 서버의 부하를 분산시킬 수 있다는 장점이 있다.

## 대표도

## 도5

## 영세서

## 도면의 간단한 설명

도 1은 종래의 실시예에 따라 웹-애플리케이션 서버들이 클라이언트들에게 컨텐츠를 제공하도록 하기 위한 시스템 구성에 대한 예시도.

도 2는 종래의 실시예에 따른 웹-애플리케이션 서버의 개략적인 블록도.

도 3은 본 발명의 실시예에 따라 웹-애플리케이션 서버들이 클라이언트들에게 컨텐츠를 제공하도록 하기 위한 시스템 구성에 대한 예시도.

도 4는 본 발명의 실시예에 따라 웹-애플리케이션 서버들이 클라이언트들의 동적 컨텐츠 요구를 분산 처리하는 과정을 개략적으로 도시한 도면.

도 5는 본 발명의 실시예에 따른 웹-애플리케이션 서버의 개략적인 블록도.

도 5a는 메타데이터의 데이터 베이스 구조를 예시적으로 도시한 도면.

도 6은 본 발명의 실시예에 따라 클라이언트로부터 동적 컨텐츠를 요청받은 근원서버의 처리 과정을 나타

낸 처리 흐름도.

도 6a는 본 발명의 실시예에 따라 근원서버로부터 대상서버로 지정된 웹-어플리케이션 서버의 처리 과정을 나타낸 처리 흐름도.

◆ 도면의 주요 부분에 대한 부호의 설명 ◆

100 : 클라이언트	200 : 인터넷
300 : 근원서버	300a, 300b, 300c : 대상서버
301 : 요청 처리기	303 : 웹-어플리케이션부
305 : 전송 매니저	307 : 시스템 모니터
309 : 컨버터	311 : 패키징/언패키징 부
313 : 송/수신부	400 : 백엔드 서버
500 : 메타데이터	

발명의 상세한 설명

발명의 목적

발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 웹-어플리케이션 서버 및 그의 동적 컨텐츠 제공 방법에 관한 것으로서, 특히, 클라이언트의 동적 컨텐츠 요구를 효율적으로 처리하기 위해, 해당 동적 컨텐츠를 생성하기 위한 웹-어플리케이션을 해당 클라이언트와 근접한 대상서버에게 전송한 후, 그 수행 결과를 해당 클라이언트에게 제공하도록 하는 것을 특징으로 하는 웹-어플리케이션 서버 및 그의 동적 컨텐츠 제공 방법에 관한 것이다.

웹에서 다양한 서비스를 제공하기 위해, 웹-서버들은 정적 및 동적 컨텐츠를 생성하여 클라이언트들에게 제공한다. 여기서 '정적 컨텐츠'는 클라이언트의 요청이 오기 전에 미리 생성되어 디스크와 같은 기록매체에 저장되어 있는 문서를 말하고, '동적 컨텐츠'는 클라이언트의 요청이 왔을 때 이를 기반으로 생성되는 문서를 말한다. 이 때, 동적 컨텐츠는 일반적으로 클라이언트의 입력 데이터나 쿠키(Cookie)를 전달받아서 생성된다. 이와 같이 정적 및 동적 컨텐츠를 서비스하는 서버를 웹-어플리케이션 서버(Web-application server)라고 한다.

도 1은 종래의 실시예에 따라 웹-어플리케이션 서버들이 클라이언트들에게 컨텐츠를 제공하도록 하기 위한 시스템 구성에 대한 예시도이다. 도 1을 참조하면, 클라이언트(10)가 인터넷(20)을 통해 웹-어플리케이션 서버(WAS: Web Application Server)(30)에게 컨텐츠를 요청하면, 웹-어플리케이션 서버(30)는 파일 또는 데이터베이스 서버와 같은 백-엔드서버(back-end server)(40)로부터 받은 데이터를 기반으로 컨텐츠를 생성하여 클라이언트(10)에게 전달한다.

도 2는 종래의 실시예에 따른 웹-어플리케이션 서버(30)의 개략적인 블록도이다. 도 1 및 도 2를 참조하면, 종래의 실시예에 따른 웹-어플리케이션 서버(30)는 요청처리기(Request Processor)(31)와, 다수의 웹-어플리케이션들(웹-어플리케이션1, ..., 웹-어플리케이션N)(33)을 포함하여 구성된다. 이들 구성에 의한 웹-어플리케이션 서버(30)의 동작을 살펴보면 다음과 같다. 먼저, 클라이언트(10)로부터 컨텐츠 요청이 발생할 경우, 요청처리기(31)는 그 클라이언트(10)와의 접속을 설정한 후, 클라이언트(10)가 요청한 컨텐츠가 정적컨텐츠인 경우 백-엔드 서버(40)로부터 받은 컨텐츠를 바로 클라이언트(10)에게 전송하고, 클라이언트(10)가 요청한 컨텐츠가 동적 컨텐츠인 경우 다수의 웹-어플리케이션들(33) 중 해당 동적 컨텐츠를 생성하기에 적절한 웹-어플리케이션을 호출한다.

그러면, 요청처리기(31)에 의해 호출된 웹-어플리케이션은 클라이언트(10)로부터 전달되어진 데이터와 백-엔드 서버(40)로부터 받은 데이터를 기반으로 컨텐츠를 생성하여 요청처리기(31)를 통해 클라이언트(10)에게 전송한다.

일반적으로 웹 서비스의 성능을 저하시키는 주된 요인들은 서버의 처리 용량 한계와 네트워크의 병목 현상으로서, 이로 인해, 클라이언트로부터 서버의 응답 시간이 늦어지게 된다. 이를 보완하기 위해 현재 인터넷 환경에서 사용되는 기술은 크게 2가지로 나누어 볼 수 있다.

첫 번째는 여러 대의 서버를 클러스터링하여 확장성과 높은 가용성을 제공하는 것이다. 즉, 여러 대의 서버를 한 클러스터로 묶고, 앞쪽에 로드 밸런서(load balancer)를 두어서, 클라이언트의 요청을 각 서버에 분산시키는 방법이다. 이는 클러스터에 서버의 수가 증가함에 따라 전체 처리 용량이 증가하게 되고, 클라이언트의 요청이 들어왔을 때 처리가 지연되거나 거절되는 것을 최소화한다는 특징이 있다.

두 번째는 정적 컨텐츠를 복사하여 인터넷상에 분산시켜 두고, 클라이언트가 컨텐츠를 요청하면, 해당 클라이언트와 가장 가까이 있는 서버가 해당 서비스를 제공하는 것이다. 이는 웹-서버의 부하를 줄일 수 있고, 네트워크 트래픽에 따른 서비스 지연을 감소시킬 수 있다는 특징이 있다.

하지만, 상기 두 가지 방법은 동적 컨텐츠를 효율적으로 서비스하기 위한 해결책으로는 적합하지 않다. 그 이유는 다음과 같다.

첫 번째, 동적 컨텐츠는 클라이언트들의 요청에 의해 실시간으로 생성하여야 하므로, 동적 컨텐츠를 인터넷에 분산하기는 힘들다. 따라서, 동적 컨텐츠를 웹 프록시 캐시(Web Proxy Cache)를 통해 투명하게 캐싱하더라도 컨텐츠의 활용도(utilization)가 낮기 때문에 캐싱 효용성이 떨어지게 된다.

두 번째, 현재의 서버 클러스터링 방법은 동적 컨텐츠를 제공하는 서버에게 로드 밸런싱을 제공할 수

없다. 즉, 정적 컨텐츠를 서비스하는 서버들을 클러스터링(clustering)할 때 사용되는 로드 밸런시는 단순히 서버 측의 네트워크 상태나 각 서버에 연결된 클라이언트의 수를 가지고 로드 밸런싱을 제공한다. 왜냐하면, 정적 컨텐츠를 제공하는 서버는 각 세션이 시간당 서버에게 주는 비교적 부하가 일정하다고 가정하기 때문이다. 하지만, 동적 컨텐츠의 경우, 서버와의 연결에서 호출되는 웹-애플리케이션이 서버에 주는 부하의 변화가 크다.

결국, 현재 통용되는 기술로서는 동적 컨텐츠를 보다 효율적으로 제공하는 데에 제약이 많다는 문제점이 있다.

#### **발명이 이루고자 하는 기술적 과제**

따라서, 본 발명에서는 이와 같은 문제점을 해결하기 위해, 동적 컨텐츠를 생성하기 위한 웹-애플리케이션을 해당 클라이언트와 근접한 대상서버에게 전송한 후, 그 수행 결과를 해당 클라이언트에게 제공하도록 하는 것을 특징으로 하는 웹-애플리케이션 서버 및 그의 동적 컨텐츠 제공 방법을 제공하는 것을 목적으로 한다.

#### **발명의 구성 및 작용**

상기 목적을 달성하기 위해, 본 발명에서 제공하는 웹-애플리케이션 서버는 클라이언트들이 요청한 컨텐츠를 해당 클라이언트에게 제공하는 웹-애플리케이션 서버에 있어서, 클라이언트가 컨텐츠를 요청할 경우, 그 컨텐츠의 종류에 따른 처리를 수행하여 해당 클라이언트에게 응답 메시지를 전송하는 요청처리기와, 상기 웹-애플리케이션 서버가 클라이언트들로부터 요청된 동적 컨텐츠를 자체적으로 생성할 수 있는지의 여부를 판단하기 위해, 상기 웹-애플리케이션 서버의 시스템 정보를 모니터링하는 시스템 모니터와, 동적 컨텐츠들을 생성하기 위한 웹-애플리케이션을 내장하고, 클라이언트가 요청한 동적 컨텐츠를 생성하는 웹-애플리케이션부와, 상기 시스템 모니터의 모니터링 결과, 상기 웹-애플리케이션 서버가 클라이언트들로부터 요청된 동적 컨텐츠를 자체적으로 생성할 수 없다고 판단된 경우, 다수의 웹-애플리케이션 서버들 각각의 정보를 관리하는 메타데이터 또는 해당 동적 컨텐츠를 요청한 클라이언트의 정보에 의거하여, 다수의 웹-애플리케이션 서버들 중에서 현재의 동적 컨텐츠 요청을 처리하는 대상서버를 선정하는 전송매니저와, 상기 전송매니저에 의해 대상서버로 선정된 웹-애플리케이션 서버와의 데이터 교환을 수행하는 송/수신부를 포함하는 것을 특징으로 한다.

한편, 상기 목적을 달성하기 위해, 본 발명에서 제공하는 웹-애플리케이션 서버의 동적 컨텐츠 제공 방법은 네트워크로 연결된 다수의 웹-애플리케이션 서버들이 클라이언트들의 동적 컨텐츠 요청에 응답하는 웹-애플리케이션 서버의 동적 컨텐츠 제공 방법에 있어서, 클라이언트가 제1 웹-애플리케이션 서버에게 동적 컨텐츠를 요청할 경우, 상기 제1 웹-애플리케이션 서버의 시스템 상태 정보를 참조하여 상기 제1 웹-애플리케이션 서버가 해당 클라이언트의 동적 컨텐츠 요구를 수행할 수 있는지의 여부를 판단하는 제1 과정과, 상기 제1 과정의 판단 결과 상기 제1 웹-애플리케이션 서버가 해당 클라이언트의 동적 컨텐츠 요구를 수행할 수 없다고 판단된 경우, 다수의 웹-애플리케이션 서버들 각각의 정보를 관리하는 메타데이터 또는 해당 동적 컨텐츠를 요청한 클라이언트의 정보에 의거하여, 다수의 웹-애플리케이션 서버들 중에서 현재의 동적 컨텐츠 요청을 수행할 제2 웹-애플리케이션 서버를 선정하는 제2 과정과, 상기 제2 웹-애플리케이션 서버에게 상기 선정 결과를 통지하고, 그 제2 웹-애플리케이션 서버에 상기 동적 컨텐츠 생성을 위해 유효한 웹-애플리케이션들이 존재하는지를 확인하는 제3 과정과, 상기 제3 과정의 확인결과 해당 웹-애플리케이션들이 존재하는 경우, 상기 클라이언트의 동적 컨텐츠 요청을 상기 제2 웹-애플리케이션 서버에서 수행하도록 한 후, 그 수행 결과를 해당 클라이언트에게 제공하는 제4 과정을 포함하는 것을 특징으로 한다. 이 때, 상기 제3 과정의 확인 결과 상기 제2 웹-애플리케이션 서버에 해당 웹-애플리케이션들이 존재하지 않는 경우, 제1 웹-애플리케이션 서버가 해당 웹-애플리케이션들을 제2 웹-애플리케이션 서버에게 전송하는 과정을 더 포함한다.

이하, 본 발명에 따른 웹-애플리케이션 서버 및 그의 동적 컨텐츠 제공 방법에 대한 바람직한 실시예를 첨부된 도면에 의거하여 상세하게 설명한다.

도 3은 본 발명의 실시예에 따라 웹-애플리케이션 서버들이 클라이언트들에게 컨텐츠를 제공하도록 하기 위한 시스템 구성에 대한 예시도이다.

도 3을 참조하면, 본 발명의 실시예에 따른 컨텐츠 제공을 위해 다수의 웹-애플리케이션 서버(300, 300a, 300b, 300c, ...)들이 인터넷을 통해 연결되어 있고, 클라이언트들(100)은 이들 중 하나의 웹-애플리케이션 서버로부터 원하는 동적 컨텐츠를 제공받을 수가 있는 것이다.

즉, 클라이언트가 최초에 접속하여 컨텐츠를 요청하는 웹-애플리케이션 서버를 '근원서버'라고 가정하고, 클라이언트(100)가 인터넷(200)을 통해 근원서버(300)에게 임의의 컨텐츠를 요청하면, 그 근원서버(300)는 요청받은 컨텐츠가 정적 컨텐츠인지 동적 컨텐츠인지에 따라 적절한 서비스를 제공한다. 만일, 클라이언트가 정적 컨텐츠를 요청할 경우, 종래의 경우와 마찬가지로, 백엔드서버(400)에게 해당 컨텐츠를 요청하여 클라이언트에게 제공한다. 한편, 클라이언트가 동적 컨텐츠를 요청할 경우, 우선 근원서버(300)가 해당 동적 컨텐츠를 서비스할 수 있는지를 판단하는 과정을 수행한다. 그리고, 근원서버(300)가 해당 동적 컨텐츠를 서비스할 수 있다고 판단된 경우, 근원서버(300) 내에 포함된 웹-애플리케이션을 이용하여 해당 동적 컨텐츠를 생성하여 해당 클라이언트에게 제공하고, 근원서버(300)가 해당 동적 컨텐츠를 서비스할 수 없다고 판단된 경우, 근원서버(300) 및 클라이언트(100)와 인터넷(200)으로 연결된 다수의 다른 웹-애플리케이션 서버들(300a, 300b, 300c) 중 하나의 서버를 대상서버로 선정하여 그 대상서버에서 해당 동적 컨텐츠를 서비스할 수 있도록 한다.

도 4는 본 발명의 실시예에 따라 웹-애플리케이션 서버들이 클라이언트들의 동적 컨텐츠 요구를 분산 처리하는 과정을 개략적으로 도시한 도면이다.

도 4를 참조하면, 클라이언트(100)가 인터넷으로 연결된 다수의 웹-애플리케이션 서버들 중 하나의 웹-애플리케이션 서버1(300, 이하, '근원서버'라 함)측으로 동적 컨텐츠를 요청하면(S101), 그 근원서버(300)는

해당 ‘서비스의’ 가능성 여부를 탐진하여, 자체적으로 서비스가 가능한 경우, 해당 클라이언트(100)의 요청에 응답을 한다(S102). 하지만, 근원서버(300)가 해당 서비스의 가능성 여부를 탐진하여, 자체적으로 서비스하는 것이 불가능하다고 판단된 경우, 인터넷에 연결된 다수의 웹-애플리케이션 서버들 중 하나의 웹-애플리케이션 서버 즉, 웹-애플리케이션 서버2(300a)(이하, ‘대상서버’라 함)를 선정하여 해당 클라이언트(100)의 동적 컨텐츠 서비스 요구를 수행할 수 있도록 한다(S103). 이 때, 근원서버(300)는 인터넷에 연결된 다수의 서버들의 각종 정보들을 저장 관리하는 메타데이터를 참조한다.

그리고, 상기 선택된 대상서버(300a)측으로 해당 클라이언트(100)의 동적 컨텐츠 요구를 리다이렉션한다(S104). 이와 동시에 해당 클라이언트(100)에게 리다이렉션 정보를 전송한다(S105).

한편, 근원서버(300)는 대상서버(300a)에 해당 동적 컨텐츠를 생성하기 위한 웹-애플리케이션이 존재하는지의 여부를 확인한다. 만일 해당 웹-애플리케이션이 존재하는 경우, 그 웹-애플리케이션의 버전까지도 확인하여, 해당 웹-애플리케이션의 유효성을 확인하는 과정을 거친다.

상기 확인결과, 만일, 대상서버(300a)에 해당 동적 컨텐츠를 생성하기 위한 웹-애플리케이션이 없거나, 있더라도 근원서버(300)에 저장된 웹-애플리케이션 보다 버전이 낮은 경우, 근원서버(300)는 해당 웹-애플리케이션들을 대상서버(300a)측으로 전송하는 과정을 수행한다. 이 때, 근원서버(300)와 대상서버(300a)의 환경이 다를 수 있으므로, 해당 웹-애플리케이션들의 설정을 대상서버(300a)의 환경에 맞추어 주는 과정을 수행한다. 또한, 근원서버(300)는 해당 웹-애플리케이션을 대상서버(300a)로 전송할 때, 그 웹-애플리케이션을 구성하는 모든 요소들을 하나로 패키징하여 전송하는 과정을 더 수행하며, 대상서버(300a)는 근원서버(300)로부터 웹-애플리케이션이 전송된 경우, 이를 언패키징하는 과정을 더 수행한다.

이와 같이 하여 대상서버(300a)가 해당 동적 컨텐츠를 생성하기 위한 준비가 완료되면, 대상서버(300a)는 해당 웹-애플리케이션들을 가지고 클라이언트가 원하는 동적 컨텐츠를 생성한다(S106).

한편, 클라이언트(100)는 근원서버(300)로부터 전송된 리다이렉션 정보에 의거하여, 대상서버(300a)로 재접속하여(S107), 원하는 동적 컨텐츠를 대상서버(300a)로부터 제공받는다(S108).

이 때, 도 4에서는 클라이언트의 요청을 대상서버(300a)로 리다이렉션(redirection)하는 경우에 대한 예를 설명하였는데, 근원서버(300)의 시스템 상태 및 근원서버(300)와 대상서버(300a)간의 통신 트래픽 상태 등에 따라, 대상서버(300a)에서 생성된 컨텐츠를 근원서버(300)에서 다시 되돌려 받아 해당 클라이언트에게 전송하는 방법(forwarding)도 가능하다.

도 5는 본 발명의 실시예에 따른 웹-애플리케이션 서버(300)의 개략적인 블록도이다. 도 5를 참조하면, 본 발명의 실시예에 따른 웹-애플리케이션 서버(300)는 클라이언트들(100)이 요청한 컨텐츠를 해당 클라이언트(100)에게 제공하는 장치를 말하는 것으로서, 요청처리기(301), 웹-애플리케이션부(303), 전송매니저(305), 시스템 모니터(307), 컨버터(309), 패키징/언패키징부(311) 및 승/수신부(313)를 포함한다.

요청처리기(301)는 클라이언트가 컨텐츠를 요청할 경우, 그 컨텐츠의 종류에 따른 처리를 수행하여 해당 클라이언트에게 응답 메시지를 전송한다. 이 때, 요청처리기(301)는 웹-애플리케이션부(303)에서 생성된 동적 컨텐츠 또는 전송매니저(305)에서 대상서버로 선정된 웹-애플리케이션 서버의 정보를 해당 클라이언트에게 전송한다. 즉, 시스템 모니터(307)의 모니터링 결과, 해당 웹-애플리케이션 서버(300)가 자체적으로 동적 컨텐츠를 생성할 수 있다고 판단된 경우 웹-애플리케이션부(303)에서 생성된 동적 컨텐츠를 해당 클라이언트에게 제공하고, 해당 웹-애플리케이션 서버(300)가 자체적으로 동적 컨텐츠를 생성할 수 없다고 판단된 경우 전송 매니저(305)가 선정한 대상서버 정보를 전송한다.

시스템 모니터(307)는 웹-애플리케이션 서버(300)가 클라이언트들로부터 요청된 동적 컨텐츠를 자체적으로 생성할 수 있는지의 여부를 판단하기 위해, 웹-애플리케이션 서버(300)의 시스템 정보를 모니터링하여 부하를 측정한다. 예를 들어, 시스템 모니터(307)는 웹-애플리케이션 서버(300) 내의 CPU 점유율 및 부하(load)의 평균값에 대한 임계값을 설정한 후, 그 CPU 점유율 및 부하(load)의 평균값을 모니터링한다. 그리고, 상기 CPU 점유율 및 부하(load)의 평균값이 상기 설정된 임계값을 초과하는 경우, 시스템 모니터(307)는 해당 웹-애플리케이션 서버(300)가 해당 동적 컨텐츠를 자체적으로 생성할 수 없다고 판단한다.

웹-애플리케이션부(303)는 동적 컨텐츠들을 생성하기 위한 웹-애플리케이션을 내장하고, 클라이언트가 요청한 동적 컨텐츠를 생성한다.

전송매니저(305)는 시스템 모니터(307)의 모니터링 결과, 해당 웹-애플리케이션 서버(300)가 클라이언트들로부터 요청된 동적 컨텐츠를 자체적으로 생성할 수 없다고 판단된 경우, 인터넷으로 연결된 다수의 웹-애플리케이션 서버들 각각의 정보를 관리하는 메타데이터(500) 및 해당 동적 컨텐츠를 요청한 클라이언트(100)의 정보에 의거하여, 다수의 웹-애플리케이션 서버들 중에서 현재의 동적 컨텐츠 요청을 리다이렉션(redirection)하기 위한 대상서버를 선정한다. 예를 들어, 전송매니저(305)는 메타데이터(500)에 포함된 각 웹-애플리케이션 서버들 중 시스템 부하가 소정 레벨 이하인 웹-애플리케이션 서버. 또는 메타데이터(500)에 포함된 각 웹-애플리케이션 서버들의 IP 주소와 해당 클라이언트(100)의 IP주소에 의거하여 해당 클라이언트(100)와 지리적으로 가까운 위치에 설치된 웹-애플리케이션 서버를 대상서버(300a)로 선정한다. (이 때, 대상서버(300a)로 선정된 웹-애플리케이션 서버는 해당 클라이언트(100)의 요청을 수행하기 위해 필요한 각종 S/W 및 H/W 관련 리소스를 모두 포함하여야 한다.)-> 이 부분은 임민열씨의 요청에 의해 추가된 부분으로서, 여기서 말하는 ‘리소스’는 임민열씨가 설명하신대로 ‘웹-애플리케이션’과는 다른 의미로 사용된 것입니다. 하지만, 이 부분은 본 발명의 주요내용과는 좀 거리가 있는 부분으로서, 일반적인 기술 내용이라고 사료되는 바, 삭제하는 것이 바람직합니다.

한편, 전송매니저(305)는 클라이언트(100)가 요청한 동적 컨텐츠를 생성하기 위해 필요한 웹-애플리케이션 정보를 도출한 후, 대상서버(300a)로 선정된 웹-애플리케이션 서버에 해당 웹-애플리케이션들이 있는지, 만일 대상서버(300a)에 해당 웹-애플리케이션들이 있는 경우, 그 버전이 유효한지를 확인한다.

그리고, 상기 해당 동적 컨텐츠를 생성하기 위해 유효한 웹-애플리케이션들이 대상서버(300a)에 존재하지 않는 경우, 전송매니저(305)는 웹-애플리케이션부(303)로부터 해당 웹-애플리케이션들을 도출하여 승/수신

부(318)를 통해 대상서버(300a)로 전송한다.

이 때, 전송매니저(305)는 해당 동적 컨텐츠를 생성하기 위해 유효한 웹-애플리케이션들이 대상서버(300a)에 존재하는지의 여부를 확인하기 위해, 메타데이터(500)에 포함된 웹-애플리케이션 서버별 웹-애플리케이션 정보를 참조하거나, 송/수신부(313)를 통해 대상서버(300a)에게 해당 웹-애플리케이션의 존재 여부를 직접 질의하는 방법을 사용한다.

또한, 전송매니저(305)는 상기 기술된 기능들을 수행하기 위해, 메타데이터(500)에 접근하여 이들을 운용하게 되는데, 전송매니저(305)가 메타데이터(500)를 운용하는 방법은 현재 다양한 방법이 가능하며, 본 발명에서는 전송매니저(305)가 메타데이터(500)를 운용하는 방법에 대해서는 한정하지 않는다.

전송매니저(305)가 메타데이터(500)를 운용하는 방법의 대표적인 예로서, LDAP(Light-weight Directory Access Protocol) 서버를 응용하는 방법이 있다. 이 경우, LDAP 서버는 메타데이터(500)를 트리(tree) 기반으로 구성하며, 그 트리(tree)의 각 노드(node)는 각 서버의 메타정보를 가지게 되는데, 스스로 자기 내용을 유지 및 변경할 수 있는 권한을 가진다. 한편, LDAP 서버는 그 노드들을 리퍼럴(referral)이라는 개념을 통해 하나의 가상 트리로서 구성하여 관리한다. 따라서, 전송매니저(305)는 단순히 이 트리를 탐색하여 원하는 정보를 얻게 되는 것이다. 한편, LDAP 서버에 의해 운용되는 별도의 메타정보 관리기를 설치하고, 메타데이터들은 그 메타정보 관리기에 의해 운용/관리되도록 하고, 전송매니저(305)는 그 메타정보 관리기로부터 메타데이터에 대한 정보를 전송받아 활용하도록 하는 방법도 실시할 수 있을 것이다.

송/수신부(313)는 전송매니저(305)에 의해 대상서버(300a)로 선정된 웹-애플리케이션 서버와의 데이터 교환을 수행한다.

컨버터(309)는 대상서버(300a)에 해당 동적 컨텐츠를 생성하기 위한 웹-애플리케이션이 존재하지 않는 경우, 웹-애플리케이션부(303)로부터 도출된 웹-애플리케이션들의 설정을 대상서버(300a)의 환경에 맞도록 변환하여 준다.

패키징/언패키징부(311)는 컨버터(309)를 통해 설정이 변환된 웹-애플리케이션들을 하나로 패키징하여 송/수신부(313)로 전송하거나, 외부의 다른 웹-애플리케이션 서버로부터 패키징되어 전송되는 웹-애플리케이션들을 언패키징하여 컨버터(309)로 전송하는 역할을 한다.

이 때, 컨버터(309) 및 패키징/언패키징부(311)는 근원서버(300) 및 대상서버(300a)의 시스템 사양에 의해 선택적으로 적용가능하다. 예를 들어, 근원서버(300)와 대상서버(300a)의 시스템 사양이 동일한 경우, 컨버터(309) 및 패키징/언패키징부(311)는 적용하지 않을 수도 있는 것이다.

이와 같은 구성을 갖는 웹-애플리케이션 서버(300)의 동작을 설명하면 다음과 같다. 우선, 클라이언트(100)가 컨텐츠를 요청하면, 요청처리기(301)는 요청된 컨텐츠의 종류를 구분하여, 정적 컨텐츠인 경우 백엔드 서버(400)로부터 필요한 컨텐츠를 받아서 해당 클라이언트(100)에게 서비스한다. 하지만, 클라이언트로부터 요청된 컨텐츠가 동적 컨텐츠인 경우, 요청처리기(301)는 시스템 모니터(307)의 모니터링 결과에 의거하여 직접 서비스할 것인지 아니면 다른 서버(대상서버)로 리다이렉션(redirection)할 것인지를 결정한다. 이 때, 만일, 직접 서비스를 제공할 경우, 요청처리기(301)는 웹-애플리케이션부(303)로부터 요청된 동적 컨텐츠를 생성하는 웹-애플리케이션을 호출하여 해당 동적 컨텐츠를 생성하여 클라이언트(100)에게 제공한다. 한편, 다른 서버로 해당 클라이언트(100)의 요청을 리다이렉션하고자 하는 경우, 전송매니저(305)는 실제로 서비스를 제공할 서버(대상서버)(300a)를 결정한 뒤, 필요한 웹-애플리케이션의 존재여부를 메타데이터(500)로부터 얻는다. 만일 대상서버(300a)에 필요한 웹-애플리케이션이 존재하는 경우, 이를 이용하여 대상서버(300a)에서 해당 컨텐츠를 생성하면 되지만, 그렇지 않은 경우, 필요한 웹-애플리케이션을 웹-애플리케이션부(303)로부터 대상서버로 전송하여야 한다. 이 때, 해당 웹-애플리케이션들은 컨버터(309)를 거쳐 대상서버의 환경에 맞도록 설정이 변경되며, 패키징/언패키징부(311)를 거쳐 하나로 패키징된다. 이와 같이 패키징된 웹-애플리케이션들은 송/수신부(313)를 통해 대상서버(300a)로 전송된다.

이 때, 전송매니저(305)는 대상서버(300a)로 선정된 웹-애플리케이션 서버의 정보를 요청처리기(301)로 전달하며, 요청처리기(301)는 이를 해당 클라이언트(100)에게 전달함으로써, 해당 클라이언트(100)가 대상서버(300a)로 재접속하여 원하는 컨텐츠를 제공받을 수 있도록 한다.

도 5a는 메타데이터(500)의 데이터 베이스 구조를 예시적으로 도시한 도면이다.

메타데이터(500)는 일반적으로 인터넷에 존재하는 각종 서버들의 정보를 관리하기 위한 데이터로서, 그 필드 구조는 다양하게 구성하는 것이 가능하다.

도 5a의 예에서는 해당 서버의 이름을 관리하는 서버\_이름\_필드(510)와, 해당 서버에 포함된 웹-애플리케이션 리스트를 관리하는 웹-애플리케이션\_리스트\_필드(520)와, 해당 서버의 시스템 로드 상태를 관리하는 서버\_로드\_필드(530)와, 해당 서버가 포함된 도메인 네임을 관리하는 도메인\_네임\_필드(540)와, 해당 서버의 IP주소를 관리하는 IP\_주소\_필드(550)를 포함하는 경우를 나타내었다.

이 때, 도 5a에는 도시되지 않았지만, 메타데이터(500)에서 각 서버에 포함된 H/W 및 S/W 목록들을 관리하도록 하여, 전송 매니저(305)가 해당 서버에 포함된 H/W 및 S/W 목록에 의거하여 서버들의 성능을 예측한 후, 그 결과에 의해 성능이 우수한 서버를 대상 서버로 선정하도록 할 수도 있다.

도 6은 본 발명의 실시예에 따라 클라이언트로부터 동적 컨텐츠를 요청받은 근원서버의 처리 과정을 나타낸 처리 흐름도이다.

도 6을 참조하면, 웹-애플리케이션 서버(이하, '근원서버'라 함)가 클라이언트로부터 동적 컨텐츠를 요청받으면(S201), 그 근원서버는 자체 시스템 상태 정보를 참조하여(S202) 해당 클라이언트의 동적 컨텐츠 요구에 대한 리다이렉션 필요성 여부를 판단한다(S203). 즉, 근원서버 내의 CPU 점유율 및 부하(load)의 평균값에 대한 임계값을 설정한 후, 그 CPU 점유율 및 부하(load)의 평균값을 모니터링하여 상기 CPU 점유율 및 부하(load)의 평균값이 그 설정된 임계값을 초과하는 경우, 리다이렉션(redirection)이 필요하다고 판단된다.

이 때, 상기 판단(S203) 결과, 리다이렉션(redirection)이 불필요하다고 판단되면, 근원서버에서 해당 클라이언트로부터 요청된 동적 컨텐츠를 생성한 후(S212), 그 동적 컨텐츠를 해당 클라이언트에게 제공한다(S213).

반면, 상기 판단(S203) 결과, 리다이렉션(redirection)이 필요하다고 판단된 경우, 외부의 메타데이터 및 해당 동적 컨텐츠를 요청한 클라이언트의 정보에 의거하여, 인터넷으로 연결된 다수의 웹-애플리케이션 서버들 중에서 현재의 동적 컨텐츠 요청을 리다이렉션하기 위한 대상서버를 선정한 후, 그 대상서버에게 리다이렉션 대상서버로 선정되었음을 통지한다(S204). 이 때, 메타데이터란 인터넷으로 연결된 다수의 웹-애플리케이션 서버들 각각의 정보를 관리하는 데이터 베이스의 일종이다. 한편, 대상서버의 선정 기준은 해당 동적 컨텐츠를 요청한 클라이언트와의 지리적인 거리 또는 시스템 부하 상태로서, 해당 클라이언트와 지리적으로 가깝고, 시스템 부하가 적은 서버를 대상서버로 선정한다.

이와 같이 대상서버가 선정되었으면, 그 대상서버에 해당 동적 컨텐츠 생성을 위해 유효한 웹-애플리케이션들이 존재하는지를 확인한다(S205). 즉, 해당 동적 컨텐츠를 생성하기 위해 적절한 버전의 웹-애플리케이션들이 존재하는지를 확인한다.

이 때, 이를 확인하는 방법은 메타데이터에 포함된 웹-애플리케이션 서버별 웹-애플리케이션 정보를 참조하는 방법과, 대상서버에게 해당 웹-애플리케이션들이 존재하는지의 여부를 직접적으로 질의하는 방법이 있을 수 있다.

상기 확인(S205) 결과, 해당 웹-애플리케이션들이 대상서버에 존재하는 경우, 클라이언트의 동적 컨텐츠 요청을 대상서버로 리다이렉션한 후(S210). 해당 클라이언트에게 그 리다이렉션 정보를 전송한다(S211).

만일, 상기 확인(S205) 결과, 해당 웹-애플리케이션들이 대상서버에 존재하지 않는 경우 근원서버에서 해당 웹-애플리케이션을 검색하고(S206), 그 웹-애플리케이션들의 설정을 대상서버의 환경에 맞도록 변환한 후(S207), 변환된 웹-애플리케이션들을 패키징하여 대상서버로 전송하는 과정(S208, S209)을 수행한 후, 상기 과정(S210, S211)을 수행한다.

이 때, 해당 웹-애플리케이션들의 설정을 대상서버의 환경에 맞도록 변환하는 과정(S207)은 상기와 같이 근원서버에서 수행하는 방법에 대하여 설명하였지만, 근원서버에서는 일단 해당 웹-애플리케이션들을 패키징만하여 전송하고, 이를 수신한 대상서버측에서 그 웹-애플리케이션들의 설정을 변환하는 방법도 가능하다.

도 6a는 본 발명의 실시예에 따라 근원서버로부터 대상서버로 지정된 웹-애플리케이션 서버의 처리 과정을 나타낸 처리 흐름도이다. 도 6a를 참조하면, 도 6에 도시된 바와 같은 처리 과정에 의해 대상서버로 선정된 웹-애플리케이션 서버가 근원서버로부터 대상서버로 선정되었음을 통보 받은 후(S301), 근원서버로부터 해당 동적 컨텐츠를 생성하기 위한 웹-애플리케이션들을 패키징한 정보를 수신하면(S302). 그 패키징된 웹-애플리케이션 정보를 언패키징한다(S303). 그리고, 언패키징에 의해 얻어진 웹-애플리케이션들에 의해 해당 동적 컨텐츠를 생성하여(S304). 대상서버에 접속하는 클라이언트에게 그 동적 컨텐츠를 제공한다(S305, S306).

이 때, 만일, 근원서버가 해당 동적 컨텐츠 생성을 위해 필요한 웹-애플리케이션들이 있는지를 확인하기 위한 질의를 할 경우, 그에 대한 응답을 하는 과정을 더 포함하며, 만일, 해당 동적 컨텐츠 생성을 위해 필요한 웹-애플리케이션들이 대상서버에 이미 존재하는 경우에는 상기 과정(S302, S303)은 생략 가능하다.

한편, 상기 도 6 및 도 6a에서는 클라이언트의 요청을 대상서버로 리다이렉션(redirection)하는 경우에 대한 예를 설명하였는데, 근원서버의 시스템 상태 및 근원서버와 대상서버간의 통신 트래픽 상태 등에 따라, 대상서버에서 생성된 컨텐츠를 근원서버에서 다시 되돌려 받아 해당 클라이언트에게 전송하는 방법(forwarding)도 가능하다.

이상의 설명은 하나의 실시예를 설명한 것에 불과한 것으로서, 본 발명은 상술한 실시예에 한정되지 않으며 첨부한 특허청구범위 내에서 다양하게 변경 가능하다. 예를 들어 본 발명의 실시예에 구체적으로 나타난 각 구성 요소의 형상 및 구조는 변형하여 실시할 수 있다.

### 본 발명의 효과

상기와 같은 본 발명의 웹-애플리케이션 서버 및 그의 동적 컨텐츠 제공 방법은 웹 서비스 품질 측면에서 다음과 같은 효과가 있다.

첫째, 웹 서비스의 확장성 문제를 해결할 수 있고, 동적으로 부하를 분산시켜 서비스의 가용성을 극대화할 수 있다.

둘째, 인터넷 트래픽 양을 감소시킨다. 즉, 현재 인터넷 트래픽의 많은 부분을 웹 트래픽이 차지하고 있는데, 클라이언트가 자신과 인접한 위치의 서버로부터 웹 서비스를 받도록 함으로써, 웹 트래픽을 크게 감소 시킬 수가 있는 것이다.

셋째, 응답시간이 획기적으로 단축된다. 즉, 대부분의 웹 서비스를 상대적으로 로드가 작은 서버 또는 클라이언트와 인접한 위치에 있는 서버에서 수행하도록 함으로써, 클라이언트의 요구에 대한 응답시간이 크게 단축된다.

또한, 본 발명은 웹 서비스 관리 측면에서 다음과 같은 효과가 있다.

첫째, 서비스 에러가 발생했을 경우 신속한 회복력을 제공할 수 있다.

둘째, 여러 서버에 분산되어 있는 웹-애플리케이션을 쉽게 통합하여 관리할 수 있다. 즉, 근원서버의 웹-애플리케이션이 수정되었을 때, 다른 서버들의 웹-애플리케이션들이 자동적으로 변경되도록 할 수 있다.

특히, 본 발명의 웹-애플리케이션 서버를 서버 관리 전문 업체와 서비스 제공자가 미리 계약을 맺어 동적

으로 변하는 클라이언트들에게 효율적으로 서비스를 제공하도록 할 경우, 서비스 제공자는 시스템을 효율적으로 유지할 수 있다는 장점이 있다.

#### (57) 청구의 범위

##### 청구항 1

클라이언트들이 요청한 컨텐츠를 해당 클라이언트에게 제공하는 웹-애플리케이션 서버에 있어서,

클라이언트가 컨텐츠를 요청할 경우, 그 컨텐츠의 종류에 따른 처리를 수행하여 해당 클라이언트에게 응답 메시지를 전송하는 요청처리기와.

상기 웹-애플리케이션 서버가 클라이언트들로부터 요청된 동적 컨텐츠를 자체적으로 생성할 수 있는지의 여부를 판단하기 위해, 상기 웹-애플리케이션 서버의 시스템 정보를 모니터링하는 시스템 모니터와.

동적 컨텐츠들을 생성하기 위한 웹-애플리케이션을 내장하고, 클라이언트가 요청한 동적 컨텐츠를 생성하는 웹-애플리케이션부와.

상기 시스템 모니터의 모니터링 결과, 상기 웹-애플리케이션 서버가 클라이언트들로부터 요청된 동적 컨텐츠를 자체적으로 생성할 수 없다고 판단된 경우, 다수의 웹-애플리케이션 서버들 각각의 정보를 관리하는 외부의 메타데이터 및 해당 동적 컨텐츠를 요청한 클라이언트의 정보에 의거하여, 다수의 웹-애플리케이션 서버들 중에서 현재의 동적 컨텐츠 요청을 리다이렉션(redirection)하기 위한 대상서버를 선정하는 전송매니저와.

상기 전송매니저에 의해 대상서버로 선정된 웹-애플리케이션 서버와의 데이터 교환을 수행하는 송/수신부를 포함하는 것을 특징으로 하는 웹-애플리케이션 서버.

##### 청구항 2

제1항에 있어서.

상기 대상서버에 해당 동적 컨텐츠를 생성하기 위해 유효한 웹-애플리케이션들이 존재하지 않는 경우, 이를 대상서버로 전송하기 위해 상기 웹-애플리케이션부에 저장된 웹-애플리케이션들의 설정을 대상서버의 환경에 맞도록 변환하는 컨버터를 더 포함하는 것을 특징으로 하는 웹-애플리케이션 서버.

##### 청구항 3

제1항 또는 제2항에 있어서.

상기 대상서버로 전송될 웹-애플리케이션들을 하나로 패키징하고, 외부의 다른 웹-애플리케이션 서버로부터 패키징되어 전송되는 웹-애플리케이션들을 언패키징하는 패키징/언패키징부를 더 포함하는 것을 특징으로 하는 웹-애플리케이션 서버.

##### 청구항 4

제1항에 있어서. 상기 요청처리기는

상기 웹-애플리케이션부에서 생성된 동적 컨텐츠 또는 상기 전송매니저에서 대상서버로 선정된 웹-애플리케이션 서버의 정보를 해당 클라이언트에게 전송하는 것을 특징으로 하는 웹-애플리케이션 서버.

##### 청구항 5

제1항에 있어서. 상기 전송매니저는

상기 메타데이터에 포함된 각 웹-애플리케이션 서버들 중 필요한 웹-애플리케이션이 존재하고, 시스템 부하가 소정 레벨 이하인 웹-애플리케이션 서버를 대상서버로 선정하는 것을 특징으로 하는 웹-애플리케이션 서버.

##### 청구항 6

제1항에 있어서. 상기 전송매니저는

상기 메타데이터에 포함된 각 웹-애플리케이션 서버들의 IP 주소 및 해당 클라이언트의 IP 주소에 의거하여, 해당 클라이언트와 지리적으로 가까운 위치에 설치된 웹-애플리케이션 서버를 대상서버로 선정하는 것을 특징으로 하는 웹-애플리케이션 서버.

##### 청구항 7

제1항에 있어서. 상기 전송매니저는

상기 대상서버로 선정된 웹-애플리케이션 서버에 상기 동적 컨텐츠를 생성하기 위해 유효한 웹-애플리케이션들이 존재하지 않는 경우, 상기 웹-애플리케이션부로부터 해당 웹-애플리케이션들을 도출하여 상기 송/수신부를 통해 상기 대상서버로 전송하는 것을 특징으로 하는 웹-애플리케이션 서버.

##### 청구항 8

네트워크으로 연결된 다수의 웹-애플리케이션 서버들이 클라이언트들의 동적 컨텐츠 요청에 응답하는 웹-애플리케이션 서버의 동적 컨텐츠 제공 방법에 있어서.

클라이언트가 제1 웹-애플리케이션 서버에게 동적 컨텐츠를 요청할 경우, 상기 제1 웹-애플리케이션 서버의 시스템 상태 정보를 참조하여 상기 제1 웹-애플리케이션 서버가 해당 클라이언트의 동적 컨텐츠 요구를

수행할 수 있는지의 여부를 판단하는 제1 과정과.

상기 제1 과정의 판단 결과 상기 제1 웹-어플리케이션 서버가 해당 클라이언트의 동적 컨텐츠 요구를 수행 할 수 없다고 판단된 경우, 다수의 웹-어플리케이션 서버들 각각의 정보를 관리하는 메타데이터 및 해당 동적 컨텐츠를 요청한 클라이언트의 정보에 의거하여, 다수의 웹-어플리케이션 서버들 중에서 현재의 동적 컨텐츠 요청을 수행할 제2 웹-어플리케이션 서버를 선정하는 제2 과정과.

상기 제2 웹-어플리케이션 서버에게 상기 선정 결과를 통지하고, 그 제2 웹-어플리케이션 서버에 상기 동적 컨텐츠 생성을 위해 유효한 웹-어플리케이션들이 존재하는지를 확인하는 제3 과정과.

상기 제3 과정의 확인 결과 해당 웹-어플리케이션들이 존재하는 경우, 상기 클라이언트의 동적 컨텐츠 요청을 상기 제2 웹-어플리케이션 서버에서 수행하도록 한 후, 그 수행 결과를 해당 클라이언트에게 제공하는 제4 과정을 포함하는 것을 특징으로 하는 웹-어플리케이션 서버의 동적 컨텐츠 제공방법.

청구항 9

제8항에 있어서, 상기 제4 과정은

상기 제3 과정의 확인 결과 해당 웹-어플리케이션들이 존재하지 않는 경우.

상기 제1 웹-어플리케이션 서버에서 해당 웹-어플리케이션들을 검색한 후, 그 웹-어플리케이션들의 설정을 상기 제2 웹-어플리케이션의 환경에 맞도록 변환하는 제4-1 과정과.

상기 제4-1 과정에서 변환된 웹-어플리케이션들을 하나로 패키징하여 상기 제2 웹-어플리케이션 서버로 전송하는 제4-2 과정을 더 포함하는 것을 특징으로 하는 웹-어플리케이션 서버의 동적 컨텐츠 제공 방법.

청구항 10

제8항에 있어서, 상기 제4 과정은

상기 제3 과정의 확인 결과 해당 웹-어플리케이션들이 존재하지 않는 경우.

상기 제1 웹-어플리케이션 서버에서 해당 웹-어플리케이션들을 검색한 후, 그 어플리케이션들을 하나로 패키징하여 상기 제2 웹-어플리케이션 서버로 전송하는 제4-1 과정과.

상기 어플리케이션들을 수신한 제2 웹-어플리케이션 서버가 그 어플리케이션들을 언패키징한 후, 해당 웹-어플리케이션들의 설정을 상기 제2 웹-어플리케이션 환경에 맞도록 변환하는 제4-2 과정을 더 포함하는 것을 특징으로 하는 웹-어플리케이션 서버의 동적 컨텐츠 제공 방법.

청구항 11

제8항 내지 제10항 중 어느 한 항에 있어서, 상기 제4 과정은

상기 제2 웹-어플리케이션 서버의 수행 결과를 상기 제1 웹-어플리케이션 서버에서 되돌려 받아 해당 클라이언트에게 제공(forwarding)하는 것을 특징으로 하는 웹-어플리케이션 서버의 동적 컨텐츠 제공 방법.

청구항 12

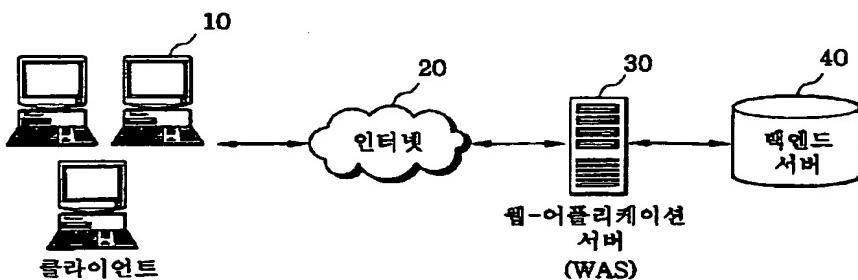
제8항 내지 제10항 중 어느 한 항에 있어서, 상기 제4 과정은

상기 클라이언트의 요청을 상기 제2 웹-어플리케이션 서버에게 리다이렉션(redirection)한 후, 상기 리다이렉션 정보를 해당 클라이언트에게 전송하는 제4-3 과정과.

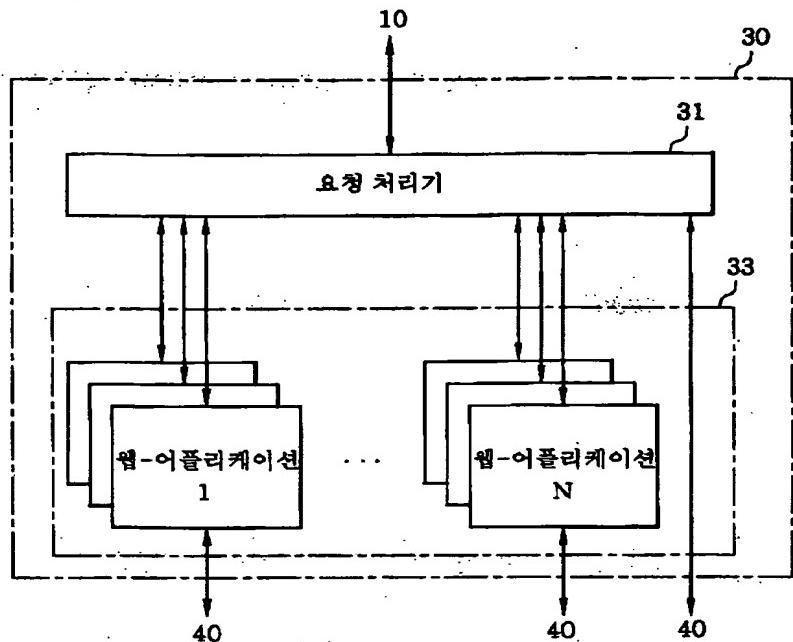
해당 클라이언트가 상기 제2 웹-어플리케이션 서버로 접속할 경우, 상기 제2 웹-어플리케이션 서버가 해당 컨텐츠를 생성하여 그 클라이언트에게 제공하는 제4-4 과정을 포함하는 것을 특징으로 하는 웹-어플리케이션 서버의 동적 컨텐츠 제공 방법.

도면

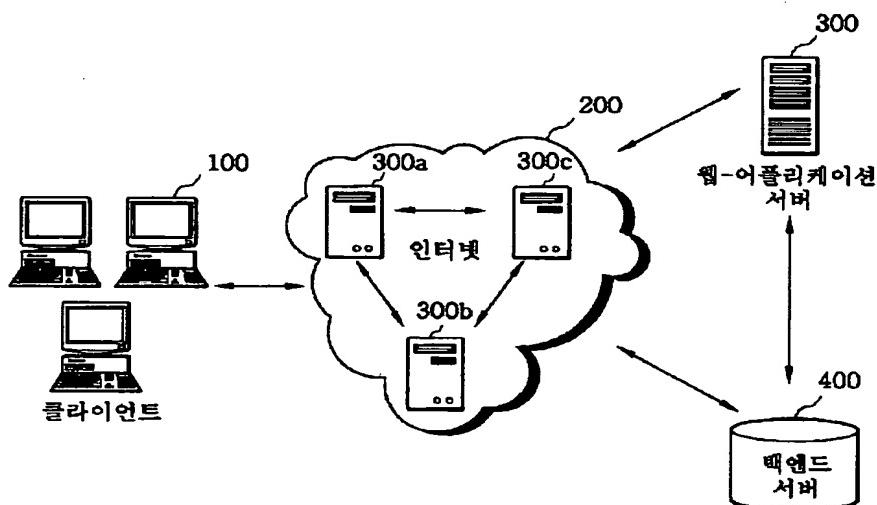
도면1



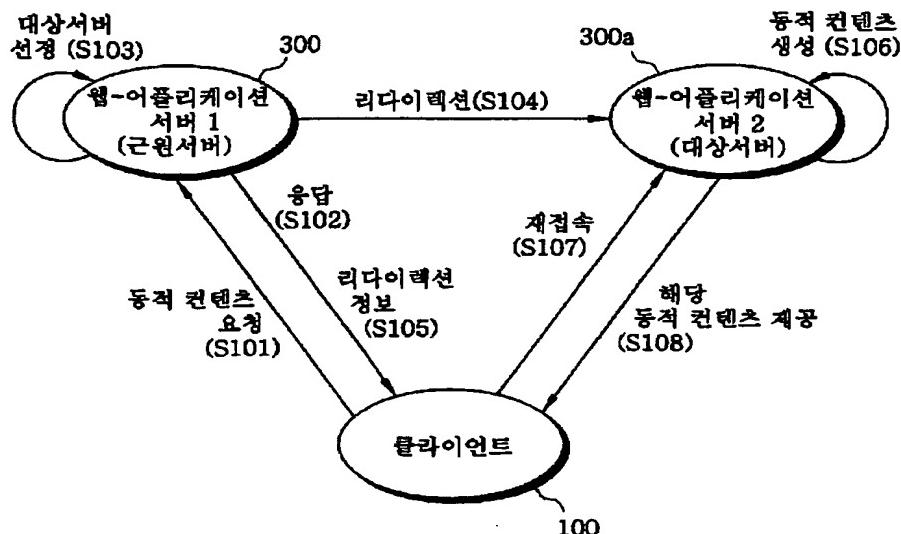
도면2



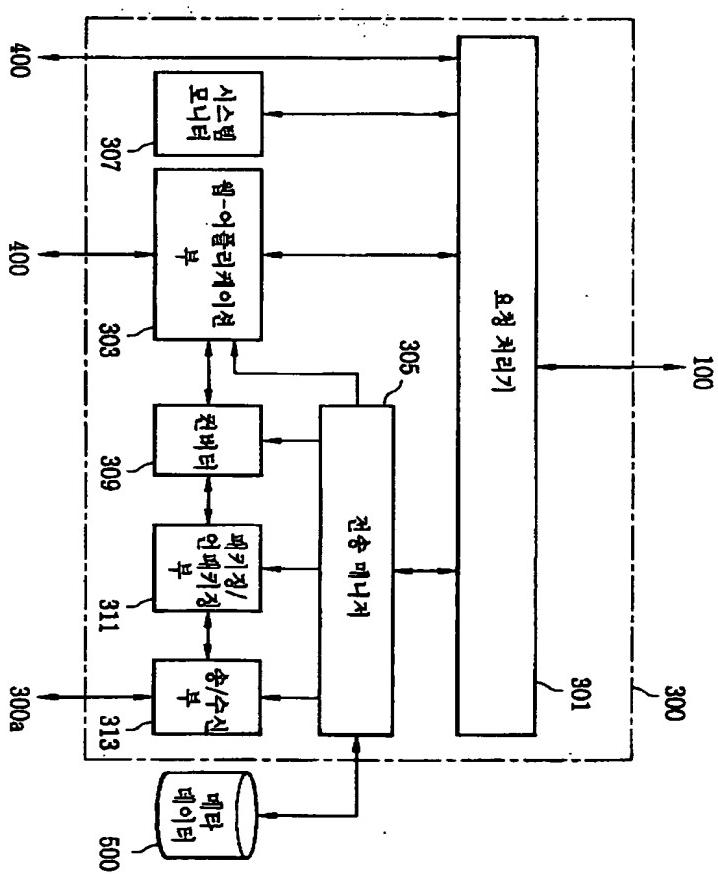
도면3



도면4



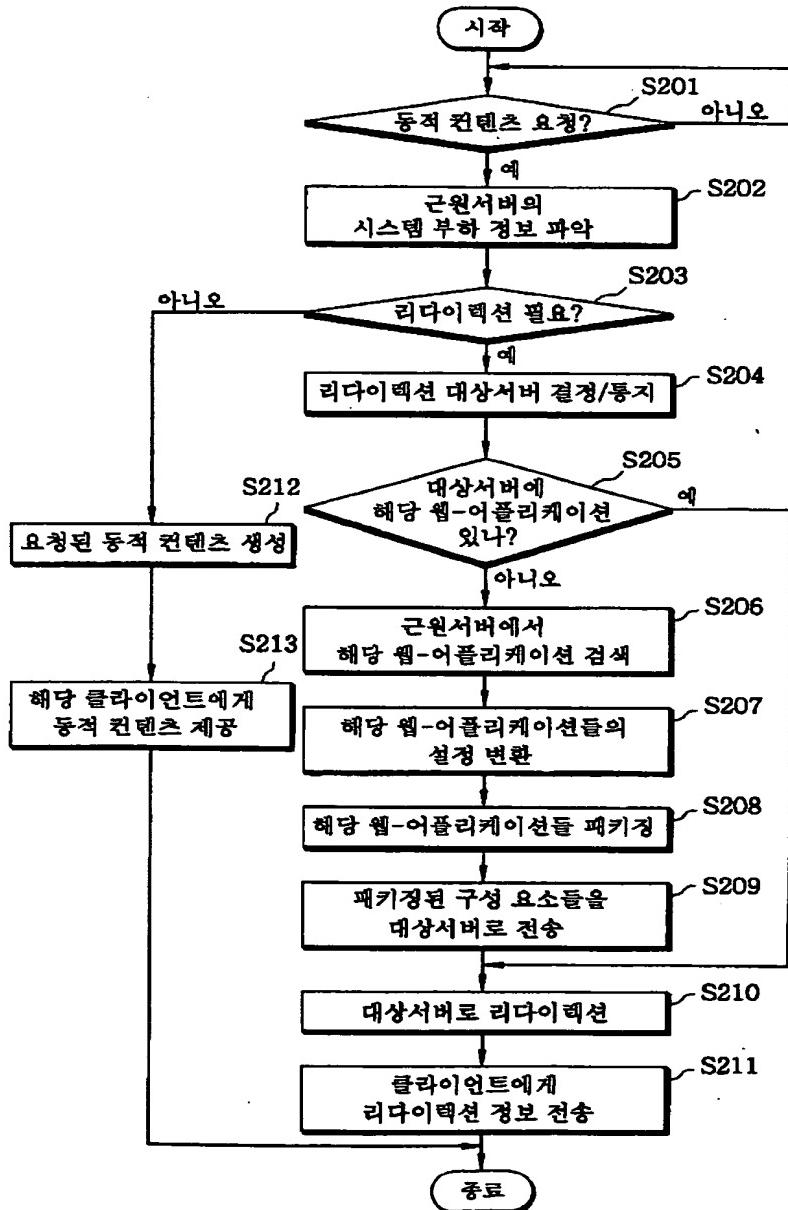
도면5



도면5a

메타 데이터 (500)	
서버 이름 (510)	
웹-어플리케이션 리스트 (520)	
서버 로드 (530)	
도메인 네임 (540)	
IP 주소 (550)	

도면6



도면6a

